**Software Implementation and Testing Document**

**For**

**Group 4: Retro-Arcade**

Version 3.0

**Authors**:

Andrew L

Joseph B

Mackenzie P

Michael T

Seth P

# Programming Languages (5 points)

Python3 – used for GUI and game design.

MySQL – Backend database to keep track of user profiles, high scores, and user stats of games played.

# Platforms, APIs, Databases, and other technologies used (5 points)

PySimpleGUI – used to create GUI in Python

Pygames – used to develop the actual games

SQLite3 – database API using MariaDB for database management

Base64 – convert string to bytes for encryption

Cryptography – API used for Fernet cipher / AES encryption on password

Datetime – used for storing / viewing the date in Highscores

~~Flask – API used to create virtual local web server to show highscores pages (in HTML)~~

# Execution-based Functional Testing (10 points)

Flappy Bird correctly tells the user prompts on starting the game as well as when the user has lost. The score for the user is displayed to the screen. User’s highest score is returned to GUI program about exiting the game. For Space Invaders, the start page, game functionality, and lose page were tested and function as expected. The user’s highest score among all of the times they wish to continue the game is returned to the GUI program upon exiting the game.

The GUI was tested to verify proper sign-up and sign-in functionality. The user’s information is stored in our database upon creating an account and is queried when the user tries to sign-in again. We tested to make sure that each time the user signs in, their game window is customized to them. We made sure when the user selects ‘My Stats’ they are able to accurately view their top five stats for each game. We further tested to make sure when the user selected Highest Scores, the all-time highest scores for each game were displayed along with the 25 users who holds that score. We also list top five players for each game. We tested the sign-out and exit functionality. We then tested to make sure every game opened when the user pressed the button associated with the game they wished to play. All CRUD functions are working for database interaction.

# Execution-based Non-Functional Testing (10 points)

Our games have not crashed after letting them run for a long period of time. Database can be queried with SQLite without issue.

# Non-Execution-based Testing (10 points)

Space Invaders’ code is clean and legible. Flappy Bird needs to be trimmed of some unnecessary commented code.

Snake’s code is clean and legible with comments. It is running as expected.

Database tables are normalized.